AUTOMATIC TEST SCENARIO GENERATOR
(ATSG)
USER'S MANUAL

20 July 1992

Prepared By: Intermetrics, Inc.
607 Louis Drive
Warminster, PA 18974

Prepared For: Naval Air Warfare Center - Aircraft Division
Warminster, PA 18974

Contract: N62269-90-C-0412

CDRL: C006

Virgil L Barnett 7/20/92
Originator Date

Stephen C. Nealy 7/20/92
Reviewer Date

AUTOMATIC TEST SCENARIO GENERATOR
(ATSG)
USER'S MANUAL

20 July 1992

Prepared By: Intermetrics, Inc.
607 Louis Drive
Warminster, PA 18974

Prepared For: Naval Air Warfare Center - Aircraft Division
Warminster, PA 18974

Contract: N62269-90-C-0412

CDRL: C006

Virgil R Barnett 7/20/92

Originator Date

Stephen C. Avalos 7/20/92

Reviewer Date

Table of Contents

1. PURPOSE

The purpose of this document is to provide the information needed to create an Automatic Test Scenario Generator (ATSG) scenario for the Navy P-3 Update IV Avionics Testbed. It provides users with the input file formats and definitions needed by the ATSG to create a deterministic test scenario. It also provides the utilities needed to preprocess the scenario file and execute the testbed.

The steps required to prepare for testbed operation from ATSG scenarios is as follows:

A. Create the ATSG input files (See Section 2)
B. Run the ATSG (Executable atsg_main.exe) (See Section 7)
C. Compile and preprocess the scenario file. (See Section 8)
D. Run the testbed and load the scenario file. (See Section 8)

2. REQUIRED INPUT FILES

The ATSG requires three input files to create the test scenario. They are the Platform Movement Input File, the Emitter Data Input File and the Event Data Input File.

When the ATSG is run, the input files must reside on the same directory as the ATSG executable: atsg main.exe.

The input files must be named as follows:

Platform Movement Input File	: platform data.dat (Section 3)
Emitter Data Input File	: emitter data.dat (Section 4)
Event Data Input File	: event data.dat (Section 5)
Miscellaneious input	: generic_input_parameters(Section 6)
	:

3. PLATFORM MOVEMENT INPUT DATA

The aircraft and each defined target are required to define at least one Initialization Record, one Line Record and one End Record. The order of the records within the input file are defined as:

Aircraft: **Initialization Record**
Line Record 1
[Arc Record 1

Line Record 2]
:
[Arc Record n
Line Record n + 1]
:
End Record.

Target i:	Initialization Record
(i=1-6)	Line Record 1
	[Arc Record 1
	Line Record 2]

:
[Arc Record n
Line Record n + 1]
:
End Record.

Record definitions can be found in the following sections. At least one space is required between elements. Separation by line is optional for appearance. For an example of the Platform Movement File, see Appendix A.

3.1 Initialization Record

The Initialization record provides the target identification, initial target position, target depth/altitude, and the initial target speed. An initialization record is required for the aircraft and each target which has data. The format of this record is:

Record Flag ID
Latitude Longitude Altitude_Depth Speed

where,

Record_Flag defines this record type and must be the 4 characters INIT.

ID is the platform ID. It is defined as 0 for the Aircraft. The range of allowable ID(s) for target platforms is an integer from 1 to Max_Number_of_Targets.

Latitude is the earth latitude of the target in degrees. The latitude must be a floating point number ranging from -90.00 to +90.00.

Longitude is the earth longitude of the target in degrees. The longitude must be a floating point number ranging from -180.00 to +180.00.

Altitude_Depth is the target depth or aircraft altitude (negative) in ft.

Speed is the platform speed in Kts.

3.2 Line Record

The Line record provides the information needed for a straight line maneuver and the terminating heading for the preceding arc record. During a line maneuver the target heading is constant but the speed of the target may be changed using an acceleration/deceleration rate. The speed of the target will not be allowed to fall below 1.0 [ft/sec] regardless of the specified deceleration rate. A Line Record must be preceded by either an Initialization Record or an Arc Record (see Section 3.3 below).

The format of this record is:

Record_Flag
Duration Heading AC_DC_Rate

where,

Record_Flag defines this record type and must be the 4 characters LINE.

Duration defines time [seconds] that the line maneuver is to execute. The duration must be non-negative. A line maneuver of duration zero may be desired to delimit two consecutive arc maneuvers. The duration must be a floating point number ranging from 0.00 to 99999.99.

Heading defines heading [degrees] of the platform during the maneuver. The heading must be a floating point number ranging from 0.00 to 359.99. This is actually the true ground course in the tactical coordinate system where (0,0) is the initial aircraft position.

AC_DC_Rate defines acceleration [positive ft/sec/sec] and deceleration [negative ft/sec/sec] of the line maneuver. A line maneuver of constant speed is created by making the rate equal to zero. The rate must be a floating point number ranging from 0.000 to 99.999.

3.3 Arc Record

The Arc record provides the information needed for an arc maneuver. During an arc maneuver the speed is constant but the platform can be made to circle left or right relative to the platform initial heading. An Arc Record must be preceded by Line Record. The format of this record is:

Record_Flag
Radius Direction Number_of_Circles

where,

Record_Flag defines this record type and must be the 3 characters ARC.

Radius is a floating point number ranging from 0.001 to 250.000 which represents the radius of the arc in Nautical Miles.

Direction defines the direction of the turn. If a left turn is desired, then the value of this item must be the 4 characters LEFT. If a right turn is desired, then the value of this item must be the 5 characters RIGHT. The change in heading is determined by the headings of the previous and following line record.

Number_of_Circles defines number of complete circles the arc maneuver is to

execute prior to stopping the maneuver at the termination heading defined by the following line record. If the number of circles is zero, the platform will proceed to the next ordered heading and end the arc maneuver. If the number of circles is "n", the platform will pass the ordered heading "n" times before ending the arc on it's "n+l" encounter. If the following line record has the same heading as the preceding line record and the Number of Circles is zero, this arc record is only a place holder connecting two line maneuvers.

3.4 End Record

The end record is provided to signal the end of maneuver data and provides the sensor sensitivity(s). These sensitivities can be set to zero for no perturbation, set to a fixed value perturbation (not random), or set to a default standard deviation perturbation based on the Update IV sensor variances. In the second case, the perturbation is fixed for guaranteeing expected results during multiple tests and multiple test cases. In the latter case, the perturbation is random based on the standard deviation. The End Record must be preceded by a Line Record.

The format for the End Record is:

Record_Flag
MAN_RDR_SS
TWS_RDR_SS
ESM_SS
MAD_SS
ACOUSTIC_SS
IRDS_SS
VISUAL_SS
COMM_SS

where,

Record_Flag defines this record type and must be the 7 characters THE_END.

MAN_RDR_SS defines the Manual RADAR perturbation fixed error value in feet. When this floating point number is a zero, it is used to make the perturbed values identical to the true values. When it is negative one (-1.0), the default perturbation standard deviation will be used.

TWS_RDR_SS defines the TWS RADAR perturbation fixed error value in feet. When this floating point number is a zero, it is used to make the perturbed

values identical to the true values. When it is negative one (-1.0), the default perturbation standard deviation will be used.

ESM_SS defines the ESM perturbation fixed error value in feet. When this floating point number is a zero, it is used to make the perturbed values identical to the true values. When it is negative one (-1.0), the default perturbation standard deviation will be used.

MAD_SS defines the MAD perturbation fixed error value in feet. When this floating point number is a zero, it is used to make the perturbed values identical to the true values. When it is negative one (-1.0), the default perturbation standard deviation will be used.

ACOUSTIC_SS defines the Acoustic perturbation fixed error value in feet. When this floating point number is a zero, it is used to make the perturbed values identical to the true values. When it is negative one (-1.0), the default perturbation standard deviation will be used.

IRDS_SS defines the IRDS perturbation fixed error value in feet. When this floating point number is a zero, it is used to make the perturbed values identical to the true values. When it is negative one (-1.0), the default perturbation standard deviation will be used.

VISUAL_SS defines the Visual perturbation fixed error value in feet. When this floating point number is a zero, it is used to make the perturbed values identical to the true values. When it is negative one (-1.0), the default perturbation standard deviation will be used.

COMM_SS defines the COMM perturbation fixed error value in feet. When this floating point number is a zero, it is used to make the perturbed values identical to the true values. When it is negative one (-1.0), the default perturbation standard deviation will be used.

4. EMITTER DATA FILE

The Emitter Data file contains 1 to Max Number of Targets Target Information Records. One record must be present for each target used in the scenario even if that target has no emitters. The order of the records within the input file is defined as:

Target 1 Information Record
Target 2 Information Record
:

Target n Information Record

where **n** can be as high as Max_Number_of_Targets. See Appendix B for an example of the Emitter Data File. Each Target Information Record is on a separate line. A separate Target Information Record must be provided for each target with events listed in the Event Data Input File (Item 5 below.) This file is effectively ended by a zero for the Target ID or an end-of-file (EOF) mark .

4.1 Target Information Record

The target information record provides the target data needed for the Emitter Data Table. The record (line) format is as follows :

Target_ID Number_of_Emitters Class Type Platform

where,

Target_ID defines the number of the target and must be an integer between 0 and Max_Number_of_Targets. Zero is used to terminate the data.

Number_of_Emitters defines the number of possible emitters on the target and must be an integer between 0 and Max_Number_of_Emitters. The Max_Number_of_Emitters is currently six.

Class defines the target classification and must be one of the three character strings Unknown, Hostile, or Friendly.

Type defines the target type and must be one of the four character strings Unknown, Subsurface, Surface, or Air.

Platform_List_Type defines the category of knowledge available for the Platform (NATO) names listed in the next item. It must be one of three character strings: KNOWN, EMPTY or UNIVERSAL. If KNOWN is not specified, no platform/NATO names may appear. UNIVERSAL and EMPTY have different interpretations as follows:

UNIVERSAL can be interpreted as unknown but critical since it produces a platform score of .5 when compared against a KNOWN. EMPTY can be interpreted as unknown but not critical since it produces a platform score of 0 when compared against anything.

(If the score is positive, the objects are considered for correlation.)

Platform defines the target (NATO) name and must be a 1 to 10 character string. Any 10 characters will be accepted as a valid name. If the Platform List Type is "KNOWN" at least one name is required; otherwise, these values are

omitted. The operational software can hold up to eight names but the ATSG fills only up to three. Each of the names, when used, must be no longer than 10 characters and must be separated from each other by one blank character. The last name must terminate the line to complete the Target Information Record.

5. EVENT DATA INPUT FILE

The event file will always start with a Model Definition Record and Scenario Definition Record. These will be followed by the respective sensor Input Record(s), Operator Input Record(s), and Situation Response Record(s). See Appendix C for an example Event Data File.

The format of an event file is:

Model Definition Record
Scenario Definition Record
Deterministic Sensor Input Record 1
:
Deterministic Sensor Input Record a
:
Deterministic Operator Input Record 1
:
Deterministic Operator Input Record b
No Action Record
:
Deterministic Situation Response Record 1
:
Deterministic Situation Response Record c
No Action Record

5.1 Model Definition Record

The model definition record defines whether the record is a deterministic or probabilistic scenario. The format of this record is:

Definition

where,

Definition must be one of two possible character strings: deterministic or probabilistic. The current delivered version of the ATSG program requires this entry to be "deterministic".

5.2 Scenario Definition Record

The scenario definition record defines the duration of the desired test scenario and the maximum number of events in the scenario. The format of this record is:

Time Max Events

where,

Time is a floating point number and defines the maximum duration of the scenario in seconds.

Max Events is an integer number between 1 and 10,000 which will be used to limit the size of the scenario file in total events.

5.3 Deterministic Sensor Input Record (DSIT)

The DSIT record is used to specify target contacts on specific sensors. The format of this record is:

Target ID Contact time Sensor Name Emitter Num P Q R

where,

Target_ID defines the number of the target and must be an integer between 1 and Max_Number_of_Targets.

Contact_time is a floating point number which defines the time in seconds from the beginning of the scenario that the contact is to be generated.

Sensor_Name is the name of the sensor generating the contact. The name is represented by one of the following character strings MAD, ESM, MANUAL_RADAR, TWS_RADAR, IRDS, VISUAL, COMM or ACOUSTIC.

Emitter_num defines the number of emitter contacts from the target and must be an integer between 0 and Max_Number_of_Emitters. An emitter number of zero must be used with a non-ESM contact.

P is the detection flag. If P is true, the detection is observed. If P is false, the detection is not generated.

Q is the classification error flag. If Q is true, the detection is given a conflicting

classification definition. If Q is false, the detection is given a conforming classification definition.

R is the geometric error flag. If R is true, the detection position will be greatly perturbed in such a way as to cause a geometric error. If R is false, the detection position will remain within the expected range tolerance.

5.4 Deterministic Operator Input Record (DOIT)

The DOIT record is used to specify asynchronous operator inputs. The format of this record is:

Action_code Time_stamp
Action_Comment

where,

Action_code is a character string which may have one of the following values: Potential_Lost_Contact, Lost_Contact, Initialize_Track, Update_Track, Collocation (Notices require no operator response), MOT_Contact, Enter_Contact, TACCO_Corr_N_C (Operator initiated switch hits), Split_Track, Change_Ctc_Num, Corr_Mode, Manual, Auto, Req_Corr, Trk_Hst, Assign_Class, Posit_Uncty, Xtend_Trk_Vec, Fix_Desig, Proj_Posit, Comp_Int, Gen_Trk, Assign_NATO_Name, Assign_DI, Coloc_Mod, Trk_Sel, No_Action; and No_Action (which is used to signal that there are no more operator inputs in the file).

Time_stamp is a floating point number which defines the time in seconds from the beginning of the scenario that the operator input is to be generated.

Action_Comment is a 40 character line of text used to describe for the testbed user what should be happening at the time of the operator action. The text will be limited to 40 characters. If it exceeds 40 characters, it will be truncated. This comment must reside on the line following the time stamp and no blank lines may intercede.

5.5 Deterministic Situation Response Record (DSRT)

The DSRT record is used to specify operator responses to expected system alerts and queries. The format of this record is:

Action_code Time_stamp
Action_Comment

where,

Action_code is a character string which may have one of the following values: Accept_Correl_Alert, Reject_Correl_Alert, Inhibit_Correl_Alert, Accept_Split_Track, Reject_Split_Track and No_Action (which is used to signal that there are no more Operator Situation Responses in the file).

Time_stamp is a floating point number which defines the time in seconds from the beginning of the scenario that the operator input is to be generated.

Action_Comment is a 40 character line of text used to describe for the testbed user what should be happening at the time of the operator action. The text will be limited to 40 characters. If it exceeds 40 characters, it will be truncated. This comment must reside on the line following the time stamp and no blank lines may intercede.

6. Miscellaneous Input file

Various input values may need to be defined using this generic_input_parameters file which must reside (if used) in the ATSG execution directory. If omitted, defaults will apply.

When used, this file may contain any number of lines used by the ATSG for input. Each line contains a label identifying the data and a value which can be an integer, float, logical, string, or an enumeration value. Lines can be in any order. Defaults are used in case of the file or any data is omitted.

The following data labels are in use at this time:
TBD

integer Dlevel.

Dlevel is the diagnostic level and is an integer value ranging from -1 to 20. This determines how much printout will be generated during the ATSG run. A value of 0 will cause normal diagnostic printout. A value of -1 causes prompts for the interactive diagnostic data to be omitted. Values from 1 to 20 will cause more diagnostics to be printed out to the screen, where 1 prints out slightly more diagnostics than 0 and 20 prints out very detailed diagnostics.

target plot option (True, Perturbed, None):".

This determines whether there will be a platform trajectories plot as described in section 6.2.2 below. It also determines whether the targets on the platform trajectory

plot are to be based on true positions or perturbed positions.

A/C plot delta time in seconds. It must be an Ada floating point value (no terminal decimal permitted).

The third prompt (unless "None" is selected above) **will be "Do you want to plot the aircraft ? Y/N (default N):".** If Y or y is entered, the aircraft trajectory plot will be written to the plot file.dat file and the fourth prompt will appear.

The fourth prompt will be **"Enter maximum time between A/C plot points in seconds:".** This only affects the time spacing of points in the platform trajectory plot. It does not affect the scenario file or the testbed. It must be any floating point number in excess of 1.0.

A final prompt will be "Do you want to extract target position? Y/N (default):". If Y or y is entered, the true target position will be written to the file text extract file.out.

The above data may also be entered via an input file using UNIX redirection as described in section 6.1 below.

7. BUILDING A SCENARIO FILE

After the four input files are built as described above, the ATSG executable program "atsg_main.exe" must be run. This executable must be run from a directory containing the three required data files. This will be called the ATSG execution directory.

The ATSG produces the scenario file "scenario file.a" which is used for testbed input. It also produces two text files which are useful to view the geometry of the scenario.

7.1 Trajectory table file

The trajectory table file, targ file.dat, is a sequence of trajectory position records in the following format:

T# time lat long "&"

where,

#	is the target number (0 for A/C)
time	is the time in seconds of the trajectory position (a negative sign only means that it is not the first position for a platform),
lat	is the position latitude in degrees north.
long	is the position longitude in degrees east.
&	is the target number (A for A/C)

7.2 Plottable file

The plottable (text) file, plot file.dat, is another sequence of trajectory positions records used for a platform trajectory plot. However, its positions are in nautical miles in conventional x,y format suitable for plotting. The format is as follows:

Platform record

.

.

.

Platform record

Corner record

Corner record

The platform record is in format:

T# xpos ypos "label data"

where,

#	is the target number (0 for A/C)
xpos	is the position longitude converted to nautical miles.
pos	is the position latitude converted to nautical miles.
label data	(when included), has the target number (blank for A/C) and the time in seconds.

and corner record is used twice to specify opposite corners of the plot for scaling purposes. This is in format:

AS xpos ypos " "

where,

xpos, ypos is a corner position in nautical miles.

7.3 Platform Trajectory Plotting

The purpose of platform trajectory plotting is to see what the scenario will look like

before it is run on the testbed. Before plotting, the operator must be logged into the SUN workstation geordi in the ATSG execution directory. To prepare the plot, the alias "graph_plot" is used to run the command "~banowetz/geordi/atsg/sort_grep.com" which prepares "plot_file.dat" for graphing and prints out the graph on the laser printer. This procedure only works on the geordi SUN workstation.

An example of the "plot_file.dat" and the generated plot are in Appendix D.

8. RUNNING A TESTBED

Before the operator prepares the "scenario_file" for use on the testbed, the file "scenario_file".a must be copied into the scenario file compile directory (for example: ~banowetz/geordi/atsg/testbed) where its file name should be changed so it does not get overwritten when another scenario file is generated. The file name must end with ".a". The file name should be given an informative name; that is, it should include either a number or a brief description (title) of the scenario like sf5.a, sf.tad.a, sf.mad1.a, sf.mad2.a, sf.clover.aircraft.a, sf.circle.aircraft.a, etc.. Where a number is used, a README file should provide a description of the available files.

The renamed scenario file must then be compiled and preprocessed in order to create a command (.cmd) file which the testbed also needs along with the ".a file". To support the compilation, some compilation units are required in the library. The command file "sglib.com" in the scenario file compile directory accomplishes this library preparation function. After this is completed, the scenario file is compiled by entering the command:

ada {filename}

The preprocessing is performed by:

preprocess {filename}.

This preprocessing results in a new preprocessed scenario filename which is the same as the input file name with ".a" replaced by ".cmd".

Once the scenario file has been preprocessed, use the rtb command (script) procedure to run the testbed. This script file is in "~banowetz/build" and takes no parameters. It is entirely interactive. When rtb is run, information will be displayed by the listing of README files. Listings are advanced by pressing the return bar after the "--more--" prompt. The operator will then be asked to "Select the base testbed dire: ". This refers to the full pathname to a directory which has a subdirectory

"a/build/pml" which contains the desired testbed executables. After entering the testbed directory and a carriage return (or only a carriage return for the default), the operator will be asked to "Enter d to run in debugger else CR:". Next a list of valid library links will be printed on the screen. The operator will be asked to "Select Link [default 1]:" where 1 is always the latest build. After selecting a valid library link, the testbed graphics will be displayed. When the manual entry panel (MEP) is displayed, the testbed is ready for operation.

Test bed operations include the following:

1. Selection of MEP switches using the left mouse button,
2. Hooking on CHRD objects using the center mouse button,
3. Testbed menu operations from the TEST_DRIVER window menu using the right mouse button.
4. SUN window operations on all windows except for the CHRD.
5. SUN window operations on the CHRD window.

Operations 1 through 4 are not available while in the debug command mode (">").

For further information on operating the testbed, refer to Boeing's Software User's Manual Testbed Software documentation dated 01 April 1990.

To begin a scenario, the operator must load up the desired scenario file. This is accomplished by selecting a scenario from the Load Scenario File option list on the TEST DRIVER window menu.

Selecting the Run Scenario option from the TEST_DRIVER window menu will run the scenario. The operator may additionally choose to set one breakpoint or single step through the scenario.

To run another scenario, the loading and running process must be performed. To quit, the operator must select "Quit" from the TEST_DRIVER window-menu.

8.1 Testbed Operation Using Debugger

If the debug mode is selected, the user should be aware that the testbed setup script will clean down the ada library. Therefore, the use of the debug mode in a directory where the ada library is used for other purposes should be avoided. A special testbed debug directory is recommended. The desired ada scenario files and the

corresponding compiled/ preprocessed scenario files must be copied into this debug directory. The a.make command must also be run initially to build an ada library for use by the debugger.

When using the debugger, the entry of debug commands and testbed operations is permitted ONLY in their respective mutually exclusive mode. The testbed starts in the debug command mode which is characterized by a ">" prompt in the testbed startup window. Debug commands can be entered such as setting breakpoints as described in the VADS Debug Manual. The "r" (restart run) or "g" (go) command will run the testbed and inhibit debug commands until a breakpoint is encountered. Testbed operations can be performed (when it is fully loaded) until a breakpoint is encountered and the mode reverts to debug command. A "g" command resumes the testbed operation mode.

Appendix A - platform data.dat Example

```
INIT 0
1.0          -177.0       -6076.0      2416.46
LINE
50.0         359.0        0.000000
ARC
10.0 LEFT    0
LINE
29.79565     89.0         0.0
ARC
10.0 LEFT    0
LINE
29.79565     179.0        0.0
ARC
10.0 LEFT    0
LINE
29.79565     269.0        0.0
ARC
10.0 LEFT    0
LINE
29.79565     359.0 0.0
ARC
10.0         LEFT         99
LINE
0.0          270.0        0.0
ARC
10.0         LEFT         0
LINE
0.0          0.0          0.0
ARC
10.0         LEFT
LINE
0.0          0.0          0.0
ARC
10.0         RIGHT
LINE
 0.0          0.0          0.0

INIT 2
1.1396       -176.6407    0.0          60.0
LINE
254.0        315.0 0.00   0.00
ARC
0.1          RIGHT        0
```

```
LINE
450.0      0.0        0.00
ARC
0.1        RIGHT      0
LINE
2000.0     45.0       0.00
ARC
10.0       LEFT       4
LINE
4800.0     90.0       -0.5


INIT 5
1.1396     -176.7407  0.0     60.0
LINE
254.0      45.0       0.00
ARC
0.10       LEFT       0
LINE
1000.0     0.0        0.00
ARC
10.0       RIGHT      4
LINE
20.0       180.0      0.00
ARC
10.0       LEFT       4
LINE
4800.0     90.0       -0.5
THE END
-1.0
-1.0
-1.0
-1.0
-1.0
-1.0
-1.0
-1.0
```

Appendix B - emitter data.dat Example

10	Friendly	Surface	known	US SHIPl USNUCl
30	hostile	subsurface	known	HOST SUB3 USNUC4 ANYOTH
20	Hostile	surface	known	HOST SUR2
40	friendly	surface	UNIVERSAL	
50	Unknown	surface	empty	

deterministic
222.0 155

5100.0	VISUAL	0	true false false
2101.0	MANUAL RADAR	0	true false false
5200.0	VISUAL	0	true false false
2 201.0	MANUAL RADAR	0	true false false
5 300.0	VISUAL	0	true false false
2 301.0	MANUAL RADAR	0	true false false
5 400.0	VISUAL	0	true false false
2 401.0	MANUAL RADAR	0	true false false
5 500.0	VISUAL	0	true false false
2 501.0	MANUAL RADAR	0	true false false
5 600.0	VISUAL	0	true false false
2 601.0	MANUAL RADAR	0	true false false
5700.0	VISUAL	0	true false false
2 701.0	MANUAL RADAR	0	true false false
5800.0	VISUAL	0	true false false
2 801.0	MANUAL RADAR	0	true false false
5 900.0	VISUAL	0	true false false
2 901.0	MANUAL RADAR	0	true false false
51000.0	VISUAL	0	true false false
21001.0	MANUAL RADAR	0	true false false
51100.0	VISUAL	0	true false false
21101.0	MANUAL RADAR	0	true false false
51400.0	VISUAL	0	true false false
21401.0	MANUAL RADAR	0	true false false

No Action

Accept Correl Alert 602.0
Expected 5 VISUAL/2 Manual radar
Accept Split Track 812.0
Expected Split as 2 Man radar moves east
No Action

Appendix D - Sample Platform Plot

T0	0.00000E+00	0.00000E+00 " 2"
T0	0.00000E+00	0.00000E+00
T5	1.56025E+01	7.08792E+00 "5 2"
T5	1.56025E+01	7.08792E+00
T2	2.15420E+01	7.11056E+00 "2 3"
T2	2.15420E+01	7.11056E+00
T0	-5.95703E-02	3.34960E+00
T0	-1.17188E-01	6.69919E+00
T0	-1.75781E-01	1.00488E+01
TC)	-2.34375E-01	1.33983E+01
T0	-2.92969E-01	1.67478E+01
T0	-3.50586E-01	2.00973E+01
T0	-4.10156E-01	2.34468E+01
T0	-4.67773E-01	2.67962E+01

.
.
.

T0	-2.04551E+01	3.31712E+01
T0	-2.02939E+01	2.98410E+01
T0	-1.90459E+01	2.67496E+01
T0	-1.68486E+01	2.42418E+01
T0	-1.39473E+01	2.25974E+01
T0	-1.06670E+01	2.20001E+01
T0	-7.37109E+00	2.25163E+01
T0	-4.43164E+00	2.40886E+01
T0	-2.17188E+00	2.65415E+01
T0	-8.48633E-01	2.96013E+01
T0	-6.05469E-01	3.29265E+01 " 1200"
T0	-6.05469E-01	3.29265E+01
T5	1.85977E+01	2.57815E+01"5 1200"
T5	1.859 ~ +01	2.57815E+01
T2	2.43262E+01	2.34254E+01"2 1201"
T2	2.43262E+01	2.34254E+01
T0	-6.05469E-01	3.29265E+01""
T2	2.43262E+01	2.34254E+01""
T5	1.85977E+01	2.57815E+01""
AS	-2.05264E+01	-1 .27386E+00 " "
AS	2.43262E+01	4.35787E+01""

Example D-l - plot file.dat file

Multi Sensor Correlation Performance Analysis for U.S. Navy P3 Update IV

6 January, 1993

Prepared By: Intermetrics, Inc.
607 Louis Drive
Warminster, PA 19874

Prepared For: Naval Air Warfare Center - Aircraft Division
Warminster, PA 19874

Contract: N62269-90-C-0412
CDRL: B006

Originator	**Date**
Reviewer	**Date**